FAULT ATTACKS ON PAIRING-BASED PROTOCOLS REVISITED

SANJIT CHATTERJEE, KORAY KARABINA, AND ALFRED MENEZES

ABSTRACT. Several papers have studied fault attacks on computing a pairing value $e(P, Q)$, where P is a public point and Q is a secret point. In this paper, we observe that these attacks are in fact effective only on a small number of pairing-based protocols, and that too only when the protocols are implemented with specific symmetric pairings. We demonstrate the effectiveness of the fault attacks on a public-key encryption scheme, an identity-based encryption scheme, and an oblivious transfer protocol when implemented with a symmetric pairing derived from a supersingular elliptic curve with embedding degree 2.

1. INTRODUCTION

Fault attacks were introduced in 1997 by Boneh, DeMillo and Lipton [7]. In these attacks, an adversary induces an error in a cryptographic device performing a secret-key operation. Using the incorrect output of the cryptographic operation, and possibly other publicly available data, the adversary may then be able to glean some information about the secret key.

Page and Vercauteren [25] were the first to consider fault attacks on pairing-based protocols. Their work motivated several other papers, most notably those of Whelan and Scott [35], Vercauteren [32], and Lashermes et al. [24]; see [15] for a recent survey. The cryptographic scenario considered in all these papers is the following. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear pairing, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order n. Suppose that party A has a secret point $Q \in \mathbb{G}_2$. During the execution of a cryptographic protocol, A computes $e(P, Q)$ where $P \in \mathbb{G}_1$ is some publicly known point. The adversary learns $e(P, Q)$ by some legitimate means, e.g., the pairing value itself is transmitted in a step of the protocol or is easily deduced from other quantities that are transmitted. During a subsequent execution of the protocol, the adversary induces a fault while A is computing $e(P, Q)$ and the adversary obtains the incorrect value $e'(P, Q)$. Note that the points P and Q are the same for both executions of the protocol. Depending on the nature of the fault introduced, the adversary may then be able to compute Q from $e(P, Q)$ and $e'(P, Q)$.

There are many possible variations of the scenario considered above. For example, the adversary may only be able to obtain the ratio $e(P, Q)/e'(P, Q)$ of the correct and faulty pairing values. Or, the adversary may only be able to obtain the ratio $e'(P, Q)/e''(P, Q)$ of two faulty pairing values.

A glaring omission in the aforementioned papers is an examination of the effectiveness of these fault attacks on specific pairing-based protocols. Indeed, none of these papers

Date: July 8, 2014.
Key words and phrases. fault attacks, pairing-based cryptography, supersingular elliptic curves.

provide a *single* example of a pairing-based protocol where the fault attacks are effective. Our examination of the literature reveals that there are relatively few protocols that release pairing values one of whose input points is secret and the other is public (or the ratio of two such pairing values, one or both of which are faulty). Such protocols include a public-key encryption (PKE) scheme [10], an identity-based encryption (IBE) scheme [16], and an oblivious transfer protocol [12]. These protocols were designed specifically to allow a reductionist security proof that does not invoke the random oracle assumption; such proofs are said to be in the 'standard model'.

The protocols in [10, 12, 16] all require an efficient method for embedding the message space (presumably the set of all bitstrings of some length ℓ) into the group $\mathbb{G}_T$. Furthermore, this embedding must also be efficiently reversible. However, it appears that such an embedding may not be constructible for asymmetric pairings including those derived from Barreto-Naehrig (BN) elliptic curves [5]. Instead, these protocols should be implemented using symmetric pairings $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ derived from supersingular elliptic curves where efficient embedding methods can be designed. Such supersingular elliptic curves include (i) the elliptic curves $Y^2 = X^3 - X \pm 1$ defined over $\mathbb{F}_{3^m}$ with embedding degree $k = 6$; (ii) the elliptic curves $Y^2 + Y = X^3 + X$ and $Y^2 + Y = X^3 + X + 1$ defined over $\mathbb{F}_{2^m}$ with embedding degree $k = 4$; and (iii) embedding degree $k = 2$ elliptic curves E defined over prime fields $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p + 1$. The security of elliptic curves with embedding degrees $k = 4$ and $k = 6$ have been tarnished by recent advances on the discrete logarithm problem in small-characteristic finite fields [19, 17, 4, 1, 2, 3, 18]. Thus, we will focus our attention on pairings derived from the $k = 2$ supersingular elliptic curves.

The remainder of the paper is organized as follows. In §2 we review the salient properties of the symmetric pairing derived from supersingular elliptic curves with embedding degree $k = 2$ and present an efficient embedding method. The efficiency of the fault attacks on these pairings is considered in §3. In §4 we give some examples of pairing-based protocols which succumb to the Page-Vercauteren and Whelan-Scott fault attacks. We draw our conclusions in §5.

2. The $k = 2$ supersingular pairing

Supersingular elliptic curves over prime fields with embedding degree 2 were used by Boneh and Franklin [8] to construct symmetric pairings for their famous identity-based encryption scheme. These pairings are also the only concrete examples of pairings given in the IETF RFC 5091 [9] specification for identity-based encryption. In this section, we follow the description of these pairings given in [21].

Let $p = 4n - 1$ be a prime, where n is also prime. Then it can be easily verified that the elliptic curve

$$E \ : \ Y^2 = X^3 - 3X \tag{1}$$

over $\mathbb{F}_p$ has $\#E(\mathbb{F}_p) = p + 1 = 4n$. The curve E is supersingular and has embedding degree $k = 2$. Since $p \equiv 3 \pmod 4$, we can represent the elements of $\mathbb{F}_{p^2}$ as $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/(i^2 + 1)$.

Note that if $\alpha = a + bi \in \mathbb{F}_{p^2}$, then $\alpha^p = a - bi$. A distortion map for E in the sense of Verheul [33] is

$$\Psi : (X, Y) \mapsto (-X, iY).$$

Let $\mathbb{G}$ and $\mathbb{G}_T$ denote the order-n subgroups of $E(\mathbb{F}_p)$ and $\mathbb{F}^*_{p^2}$, respectively. The (reduced) Tate pairing is a map

$$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$$

defined as

$$e(P, Q) = f_{n,P}(\Psi(Q))^{(p^2-1)/n} = f_{n,P}(\Psi(Q))^{4(p-1)},$$

where the Miller function $f_{n,P}$ is a rational function defined over $\mathbb{F}_p$ with divisor $(f_{n,P}) = n(P) - (nP) - (n-1)(\infty)$. The Miller function value $f_{n,P}(\Psi(Q))$ can be computed using Algorithm 1. The exponentiation by $4(p-1)$ is relatively inexpensive since

$$(a + bi)^{p-1} = \frac{a - bi}{a + bi}$$

for all $a + bi \in \mathbb{F}^*_{p^2}$.

Algorithm 1 Computing the Miller function value $f_{n,P}(\Psi(Q))$, where $P, Q \in \mathbb{G}$.

1: Write $n = \sum_{j=0}^{s-1} n_j 2^j$ with $n_j \in \{0, 1\}$ and $n_{s-1} = 1$
2: $T \leftarrow P$, $f \leftarrow 1$
3: **for** j **from** $s - 2$ **downto** 0 **do**
4: Let L denote the tangent line to E at T
5: $T \leftarrow 2T$
6: $f \leftarrow f^2 \cdot L(\Psi(Q))$
7: **if** $n_j = 1$ and $j \neq 0$ **then**
8: Let L denote the line through T and P
9: $T \leftarrow T + P$
10: $f \leftarrow f \cdot L(\Psi(Q))$
11: **end if**
12: **end for**
13: **return** f

Let $\mathbb{G}_\Phi$ denote the order-$(p+1)$ subgroup of $\mathbb{F}^*_{p^2}$. Now, $a + bi \in \mathbb{F}_{p^2}$ belongs to $\mathbb{G}_\Phi$ if and only if $(a + bi)^{p+1} = a^2 + b^2 = 1$. Hence, the elements of $\mathbb{G}_\Phi$ are in 1-1 correspondence with the points on the unit circle over the integers modulo p. It is well known [31] that these points are in 1-1 correspondence with the integers in the interval $[0, p]$, a bijection ρ being:

$$p \mapsto (0, -1), \quad \text{and } \lambda \mapsto \left(\frac{-2\lambda}{1 + \lambda^2}, \frac{1 - \lambda^2}{1 + \lambda^2} \right) \text{ for } \lambda \in [0, p).$$

The inverse of ρ is given by:

$$(0, -1) \mapsto p, \quad (0, 1) \mapsto 0, \quad \text{and } (a, b) \mapsto 2(b - 1)/a \text{ for } a \neq 0.$$

This suggests the following probabilistic embedding algorithm. Let t denote the bitlength of p, and let $\ell = t - 21$. The message space is $\mathcal{M} = \{0,1\}^\ell$. To encode a message $m \in \mathcal{M}$, repeatedly append randomly chosen bitstrings of length 20 to m until the resulting bitstring m' satisfies $\rho(m') \in \mathbb{G}_T$; then the embedding of m is $\rho(m')$. Membership of an element $\alpha \in \mathbb{G}_\Phi$ in $\mathbb{G}_T$ can be determined by checking whether $\alpha^n = 1$. Since $[\mathbb{G}_\Phi : \mathbb{G}_T] = 4$, we expect to perform four iterations of the procedure before $\rho(m') \in \mathbb{G}_T$. Note that m can be efficiently recovered from $\rho(m')$ by computing $\rho^{-1}(\rho(m'))$ and then discarding the 20 padding bits.

Remark 1. BN curves yield the most efficient pairings for implementing pairing-based protocols at the 128-bit security level. For these pairings, $\mathbb{G}_T$ is the order-n subgroup of $\mathbb{F}_{p^{12}}^*$ where $p = 36z^4 + 36z^3 + 24z^2 + 6z + 1$ and $n = 36z^4 + 36z^3 + 18z^2 + 6z + 1$ are 128-bit primes (for an appropriately chosen BN parameter z). The natural representation for elements of $\mathbb{F}_{p^{12}}$ is as univariate polynomials of degree less than 12 over $\mathbb{F}_p$. However, we were unable to identify any property of the subset of those polynomials belonging to $\mathbb{G}_T$ that would yield an efficiently-computable (and invertible) embedding.

Remark 2. Scott [29] described the implementation of asymmetric pairings derived from ordinary elliptic curves E over prime fields $\mathbb{F}_p$ with embedding degree $k = 2$. In these pairings, $\mathbb{G}_T$ is the order-n subgroup of $\mathbb{F}_{p^2}^*$ where the bitlength of n is significantly smaller than that of p; for example, one could select 160-bit n and 512-bit p in order to achieve the 80-bit security level. However, we were unable to construct an efficiently-computable (and invertible) embedding into $\mathbb{G}_T$.

Remark 3. Let E be an elliptic curve defined over $\mathbb{F}_q$, and let n be a prime divisor of $\#E(\mathbb{F}_q)$. Suppose that the embedding degree of E with respect to n is $k > 1$, and let $\mathbb{G}_T$ denote the order-n subgroup of $\mathbb{F}_{q^k}^*$. Then $\mathbb{G}_T$ is a proper subgroup of the algebraic torus $T_k(\mathbb{F}_q)$, where $\#T_k(\mathbb{F}_q) = \Phi_k(q) = h \cdot n$ and where $\Phi_k(X)$ denotes the k-th cyclotomic polynomial. It is known that efficiently computable and invertible embeddings into $T_k(\mathbb{F}_q)$ exist if one can contruct an efficient rational parameterization of $T_k(\mathbb{F}_q)$. In fact, it is known that $T_k(\mathbb{F}_q)$ is rational if k is a prime power, or the product of two prime powers. And, efficient rational parameterizations of $T_k(\mathbb{F}_q)$ are known for $k = 2, 6$; see [26]. However, generalizations of these embeddings from $T_k(\mathbb{F}_q)$ to proper subgroups $\mathbb{G}_T$ are not known, in particular if the cofactor h is large. This gives evidence that efficient and invertible embeddings into G_T may not exist for BN curves and for the curve in Remark 2.

3. Fault attacks on the $k = 2$ supersingular pairing

We demonstrate that the Page-Vercauteren [25] and Whelan-Scott [35] fault attacks can be successfully mounted on the $k = 2$ supersingular pairing. The former attack changes the number of iterations of the loop in Algorithm 1, for example by causing it to end prematurely. The latter attack corrupts a data item in some way.

3.1. **Page-Vercauteren fault attack.** The basic idea of the Page-Vercauteren fault attack [25] is to obtain two faulty pairing values

$$f_{n\pm d,P}(\Psi(Q))^{4(p-1)} \quad \text{and} \quad f_{n\pm d+1,P}(\Psi(Q))^{4(p-1)}$$

for some known d, and then use the ratio of these values to deduce Q.

We consider the scenario where the adversary causes the loop in Algorithm 1 to end prematurely. Recall that $n = \sum_{j=0}^{s-1} n_j 2^j$. Let $n' = \sum_{j=\ell}^{s-1} n_j 2^{j-\ell}$ for some $\ell \geq 1$. Note that the binary representation of n' can be obtained from the binary representation of n by deleting the last ℓ bits of the latter. Without loss of generality, suppose that n' is odd.

The adversary causes the loop in Algorithm 1 to terminate prematurely at the end of the iteration with $j = \ell$, thereby obtaining the faulty pairing value

$$e'(P,Q) \;=\; f_{n',P}(\Psi(Q))^{4(p-1)}.$$

In a subsequent iteration of the attack, the adversary causes the loop in Algorithm 1 to terminate prematurely after step 6 in the iteration with $j = \ell$, thereby obtaining the faulty pairing value

$$e''(P,Q) \;=\; f_{n'-1,P}(\Psi(Q))^{4(p-1)}.$$

The ratio of these pairing values is

$$R \;=\; \left(\frac{f_{n',P}(\Psi(Q))}{f_{n'-1,P}(\Psi(Q))}\right)^{4(p-1)} \;=\; (L(\Psi(Q)))^{4(p-1)} \;=\; (-aX_Q + bY_Q i + c)^{4(p-1)},$$

where $L(X,Y) = aX + bY + c$ is the equation of the line through the points $(n'-1)P$ and P, and where $Q = (X_Q, Y_Q)$. Since P, n' and R are known, the adversary's task is to determine Q given $a, b, c \in \mathbb{F}_p$ and $R \in \mathbb{F}_{p^2}$.

Noting that

$$R = \left(\frac{-aX_Q - bY_Q i + c}{-aX_Q + bY_Q i + c}\right)^4,$$

the adversary computes a fourth root T of R in $\mathbb{F}_{p^2}$; the other fourth roots are $-T$, iT and $-iT$. For each fourth root $t_1 + t_2 i$, the adversary finds $X, Y \in \mathbb{F}_p$ that satisfy

$$t_1 + t_2 i = \frac{-aX - bYi + c}{-aX + bYi + c} \tag{2}$$

and also the curve equation $Y^2 = X^3 - 3X$. This can be accomplished by clearing the denominator of (2) and then equating real parts[1] to obtain the linear equation

$$(a - at_1)X - bt_2 Y - c(1 - t_1) = 0. \tag{3}$$

Multiplying both sides of (3) by $(a - at_1)X + bt_2 Y - c(1 - t_1)$ yields

$$((a - at_1)X - c(1 - t_1))^2 - (bt_2 Y)^2 = 0.$$

Then, substituting $Y^2 = X^3 - 3X$ gives a cubic equation

$$-b^2 t_2^2 X^3 + a^2(t_1 - 1)^2 X^2 + (-2act_1^2 + 3b^2 t_2^2 + 4act_1 - 2ac)X + c^2(t_1 - 1)^2 = 0, \tag{4}$$

[1]Equating imaginary parts of (2) yields a linear equation that is linearly dependent on (3).

which can be easily solved for $X \in \mathbb{F}_p$.

Thus, for each fourth root $t_1 + t_2 i$, we obtain at most three candidate points (X, Y). If only one of these points has order n in $E(\mathbb{F}_p)$, then that point is the secret point Q. If there is more than one such point, then Q can be determined by computing the pairing value $e(P, Q)$ and comparing it with the value obtained from a legitimate run of the protocol.

We note that the adversary need only find the ratio R of the faulty pairing values $e'(P, Q)$ and $e''(P, Q)$, and not the values themselves.

Example 1. We chose $p = 2^{1502} + 3965739$, with $n = (p + 1)/4$ being a 1500-bit prime and $s = 1500$.[2] We ran 10 experiments with randomly-chosen elliptic curve points $P, Q \in \mathbb{G}$ and with $\ell = 19$. In each experiment we found, for each fourth root $t_1 + t_2 i$ of R, the points $Q_1 \in \mathbb{F}_p \times \mathbb{F}_p$ that satisfy the cubic equation (4). Among these points, we determined which points Q_2 are in $\mathbb{G}$. Finally, among the points Q_2, we determined the points Q_3 that yield the correct pairing value, i.e., $e(P, Q_3) = e(P, Q)$. Our results are summarized in the following table where $\#Q_i$ denotes the number of points Q_i for $i = 1, 2, 3$.

Experiment	$\#Q_1$	$\#Q_2$	$\#Q_3$	Experiment	$\#Q_1$	$\#Q_2$	$\#Q_3$
1	6	3	1	6	5	2	1
2	7	3	1	7	6	2	1
3	5	3	1	8	3	2	1
4	3	1	1	9	6	2	1
5	3	1	1	10	3	1	1

3.2. **Whelan-Scott fault attack.** Whelan and Scott [35] consider the situation where the adversary is able to change the sign bit of an $\mathbb{F}_p$-component of a single line function value $L(\Psi(Q))$.

For the sake of concreteness, suppose that $n_\ell = 1$ where $\ell \in [1, s-1]$, and suppose that the adversary flips the sign of the imaginary part of the line function value $L(\Psi(Q))$ in step 10 of iteration $j = \ell$ of Algorithm 1. Then the ratio of the correct pairing value and the faulty pairing value is

$$R = \left(\frac{-aX_Q + bY_Q i + c}{-aX_Q - bY_Q i + c} \right)^{2^\ell \cdot 4(p-1)},$$

where $L(X, Y) = aX + by + c$ is the equation of the line through $\overline{n}P$ and P, and where $\overline{n} = -1 + \sum_{j=\ell}^{s-1} n_j 2^{j-\ell}$ and $Q = (X_Q, Y_Q)$. Since P, $\overline{n}$ and R are known, the adversary's task is to determine Q given $a, b, c \in \mathbb{F}_p$ and $R \in \mathbb{F}_{p^2}$.

Noting that

$$R = \left(\frac{-aX_Q - bY_Q i + c}{-aX_Q + bY_Q i + c} \right)^{2^{\ell+3}}, \tag{5}$$

[2]For real-world implementations of the pairing derived from $k = 2$ supersingular elliptic curves, the prime p should be randomly chosen to avoid attacks on the discrete logarithm problem in $\mathbb{F}_{p^2}$ that exploit the low Hamming weight of p [28].

the adversary computes all the $2^{\ell+3}$-th roots of R in $\mathbb{F}_{p^2}$. For each such root $t_1 + t_2 i$, the adversary solves (2) for $X, Y \in \mathbb{F}_p$ as described in §3.1, and determines which solution (X, Y) is the secret point Q. To solve (5), one notes that $((-aX_Q - bY_Q i + c)/(-aX_Q + bY_Q i + c))^4$ is an element of $\mathbb{G}_T$. Hence, finding the $2^{\ell+3}$-th roots of R in $\mathbb{F}_{p^2}$ is equivalent to finding the fourth roots of R^m in $\mathbb{F}_{p^2}$, where m is the multiplicative inverse of $2^{\ell+1}$ modulo n.

Example 2. As in Example 1, we chose $p = 2^{1502} + 3965739$ with $n = (p+1)/4$ being a 1500-bit prime and $s = 1500$. We ran 10 experiments with the same elliptic curve points $P, Q \in \mathbb{G}$ as were chosen in Example 1, and with $\ell = 19$. In each experiment we found, for each fourth root $t_1 + t_2 i$ of R^m, the points $Q_1 \in \mathbb{F}_p \times \mathbb{F}_p$ that satisfy the cubic equation (4). Among these points, we determined which points Q_2 are in $\mathbb{G}$. Finally, among the points Q_2, we determined the points Q_3 that yield the correct pairing value, i.e., $e(P, Q_3) = e(P, Q)$. Our results are summarized in the following table where $\#Q_i$ denotes the number of points Q_i for $i = 1, 2, 3$.

Experiment	$\#Q_1$	$\#Q_2$	$\#Q_3$	Experiment	$\#Q_1$	$\#Q_2$	$\#Q_3$
1	6	3	1	6	4	1	1
2	3	3	1	7	5	2	1
3	6	3	1	8	5	1	1
4	8	3	1	9	2	2	1
5	3	2	1	10	6	2	1

4. Protocols

Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a symmetric pairing derived from a $k = 2$ supersingular elliptic curve, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order n. In this section, we shall assume that $\mathbb{G}$ is written multiplicatively.

We observe that many pairing-based protocols are not vulnerable to fault attacks whose objective is to let an adversary determine a secret point $Q \in \mathbb{G}$ by obtaining information about faulty pairing values $e'(P, Q)$. This is because these protocols generally apply a cryptographic hash function $\overline{H}$ to the pairing value $e(P, Q)$. As a consequence, the adversary is only able to obtain information about the hash $\overline{H}(e'(P, Q))$ of faulty pairing values. The application of $\overline{H}$ destroys any exploitable algebraic relationship between $e'(P, Q)$ and the correct pairing value $e(P, Q)$, rendering the fault attack ineffective.

To illustrate this, consider the basic version of the Boneh-Franklin identity-based encryption scheme [8]. Let g be a fixed generator of $\mathbb{G}$. The Private Key Generator selects a private key $t \in_R [1, n-1]$ and computes its public key $T = g^t$. A party with identifier ID is assigned the public key $Q = H(\mathrm{ID})$, where $H : \{0,1\}^* \to \mathbb{G}$ is a hash function, and is given the private key $d = Q^t$. To encrypt a message $m \in \{0,1\}^\ell$ for that party, a user selects $r \in_R [1, n-1]$ and computes $Q = H(\mathrm{ID})$, $R = g^r$, and $c = m \oplus \overline{H}(e(T, Q)^r)$; the ciphertext is $C = (R, c)$. To decrypt, the party possessing private key d computes $m = c \oplus \overline{H}(e(R, d))$. Suppose that an adversary selects m, computes $C = (R, c)$, and sends C to the decryptor. Suppose also that the adversary is able to induce a fault while the decryptor is computing

$e(R,d)$, and is subsequently able to obtain the faulty plaintext m'. The adversary can then compute $m' \oplus c = \overline{H}(e'(R,d))$ and $e(R,d) = e(T,Q)^r$. However, the adversary is apparently unable to recover d from these values.

In §4.1, §4.2 and §4.3, we show that fault attacks that target the computation of a pairing value are indeed effective on the Boyen-Mei-Waters public-key encryption scheme [10], Gentry's identity-based encryption scheme [16], and an oblivious transfer protocol [12]. These schemes were specifically designed to avoid the random oracle assumption in their reductionist security proofs, and consequently do not hash pairing values.

4.1. **Boyen-Mei-Waters public-key encryption scheme.** In [10], Boyen, Mei and Waters proposed a CCA-secure public key encryption scheme based on the Waters IBE scheme [34]. The original scheme is described in the asymmetric pairing setting. However, as we have already noted, no efficient embedding is currently known in the asymmetric pairing setting. We first present the Boyen-Mei-Waters public key encryption scheme in the symmetric pairing setting and then show that the protocol is vulnerable to fault attacks.

4.1.1. *Boyen-Mei-Waters scheme.* Let g be the fixed generator of $\mathbb{G}$. Let $H_s : \mathbb{G}_T \times \mathbb{G} \to \{0,1\}^k$ be a family of collision-resistant functions where $k \approx \lg(n)$.

KEY GENERATION. The user selects $\alpha \in_R [0, n-1]$ and sets $h = g^\alpha$ and $Z = e(g,h)$. The user selects $y' \in_R [0, n-1]$ and a random vector $\overrightarrow{y} = (y_1, \ldots, y_k)$ with entries from $[0, n-1]$ and computes $u' = g^{y'}$ and $u_i = g^{y_i}$ for $1 \le i \le k$. The user also selects a random function H_s. The private key is $(h, y', y_1, \ldots, y_k)$ and the corresponding public key is $(H_s, Z, u', u_1, \ldots, u_k)$.

ENCRYPTION. To encrypt a message $m \in \mathbb{G}_T$, the sender selects $r \in_R [0, n-1]$ and computes $C_0 = m \cdot Z^r = m \cdot e(g,h)^r$ and $C_1 = g^r$. She then derives a k-bit string $w = H_s(C_0, C_1)$ and computes $C_2 = (u' \prod_{i=1}^k u_i^{w_i})^r$ where $w_i \in \{0,1\}$ is the i-th bit in w. The ciphertext is $C = (C_0, C_1, C_2)$.

DECRYPTION. To decrypt $C = (C_0, C_1, C_2)$, the receiver first obtains $w = H_s(C_0, C_1)$ and computes $w' = y' + \sum_{i=1}^k y_i w_i \bmod n$ where w_i is the i-th bit in w. The receiver then tests whether $C_2 = C_1^{w'}$. If the test is successful then the receiver computes

$$m = C_0 \cdot e(C_1, h)^{-1}. \tag{6}$$

4.1.2. *Fault attack.* We assume that the adversary $\mathcal{A}$ can induce a Whelan-Scott sign-change fault (see §3.2) while the pairing values in (6) are being computed, and is subsequently able to obtain the decrypted message.

The attack proceeds as follows:

(i) $\mathcal{A}$ selects a plaintext message $m \in \mathbb{G}_T$ and computes the ciphertext $C = (C_0, C_1, C_2)$.
(ii) $\mathcal{A}$ sends C to the receiver.
(iii) While the receiver computes the pairing value $e(C_1, h)$ in (6), $\mathcal{A}$ induces a sign-change fault which causes the receiver to compute the faulty pairing value $e'(C_1, h)$.
(iv) $\mathcal{A}$ obtains the (faulty) decryption $m' = C_0 \cdot e'(C_1, h)^{-1}$.

(v) $\mathcal{A}$ computes

$$\frac{m'}{m} = \frac{e(C_1, h)}{e'(C_1, h)},$$

and thereafter computes h as described in §3.2. Note that, if it is needed, the adversary can compute the correct pairing value since $e(C_1, h) = C_0 \cdot m^{-1}$.

With the knowledge of h, the adversary can now decrypt any ciphertext that is encrypted under the corresponding public key. The other components of the private key, i.e., $y', y_1, \ldots, y_k$ are used during decryption only for the purpose of checking the validity of the ciphertext and are not needed to recover the plaintext.

Remark 4. The Boyen-Mei-Waters scheme also succumbs to the Page-Vercauteren fault attack described in §3.1. The adversary obtains two faulty decryptions $m' = C_0 \cdot e'(C_1, h)^{-1}$ and $m'' = C_0 \cdot e''(C_1, h)^{-1}$, and thereafter computes $m'/m'' = e''(C_1, h)/e'(C_1, h)$.

4.2. **Gentry's identity-based encryption scheme.** Gentry [16] presented an identity-based encryption scheme and a reductionist security proof that does not invoke the random oracle assumption. The scheme assumes that plaintext messages are elements of $\mathbb{G}_T$.

4.2.1. *Gentry's scheme.*

SETUP. The Private Key Generator (PKG) selects $g, h_1, h_2, h_3 \in_R \mathbb{G}$, $\alpha \in_R [0, n-1]$, and a hash function H from a family of universal one-way hash functions. It computes $g_1 = g^\alpha$. The public parameters are $(g, g_1, h_1, h_2, h_3, H)$, and the PKG's private key is α.

KEY EXTRACTION. To generate a private key for the party with identifier $\mathrm{ID} \in [0, n-1]$, the PKG selects $r_1, r_2, r_3 \in_R [0, n-1]$ and computes $d_i = (h_i g^{-r_i})^{1/(\alpha - \mathrm{ID})}$ for $i = 1, 2, 3$. The party's private key is $(r_1, r_2, r_3, d_1, d_2, d_3)$.

ENCRYPTION. To encrypt a message $m \in \mathbb{G}_T$ for the party with identifier ID, the sender selects $s \in_R [0, n-1]$ and computes $u = g_1^s g^{-s \cdot \mathrm{ID}}$, $v = e(g, g)^s$, $w = m \cdot e(g, h_1)^{-s}$, and $y = e(g, h_2)^s \cdot e(g, h_3)^{s\beta}$, where $\beta = H(u, v, w)$. The ciphertext is $C = (u, v, w, y)$.

DECRYPTION. To decrypt $C = (u, v, w, y)$, the receiver who possesses the private key $(r_1, r_2, r_3, d_1, d_2, d_3)$ corresponding to identifier ID computes $\beta = H(u, v, w)$ and tests whether $y = e(u, d_2 \cdot d_3^\beta) \cdot v^{r_2 + r_3 \beta}$. If the test is successful, then the receiver computes

$$m = w \cdot e(u, d_1) \cdot v^{r_1}. \tag{7}$$

4.2.2. *Fault attack.* We assume that the adversary $\mathcal{A}$ can induce a Whelan-Scott sign-change fault (see §3.2) while the pairing value in (7) is being computed, and is subsequently able to obtain the decrypted message. We also assume that $\mathcal{A}$ is able to mount a conventional timing [22] or simple power analysis [23] attack while the receiver is computing v^{r_1} in (7).

The attack proceeds as follows:

(i) $\mathcal{A}$ selects a plaintext message $m \in \mathbb{G}_T$ and computes the ciphertext $C = (u, v, w, y)$.
(ii) $\mathcal{A}$ sends C to the receiver.
(iii) While the receiver computes v^{r_1} in (7), $\mathcal{A}$ mounts a timing or simple power analysis attack and learns r_1.

(iv) While the receiver computes the pairing value $e(u, d_1)$ in (7), $\mathcal{A}$ induces a sign-change fault which causes the receiver to compute the faulty pairing value $e'(u, d_1)$.
(v) $\mathcal{A}$ obtains the (faulty) decryption $m' = w \cdot e'(u, d_1) \cdot v^{r_1}$.
(vi) $\mathcal{A}$ computes

$$\frac{m}{m'} = \frac{e(u, d_1)}{e'(u, d_1)},$$

and thereafter computes d_1 as described in §3.2. Note that, if it is needed, the adversary can compute the correct pairing value since $e(u, d_1) = e(g, h_1)^s \cdot v^{-r_1}$.

Now, with knowledge of r_1 and d_1, the adversary can decrypt any ciphertext that is intended for the party with identifier ID. Note that the other components r_2, r_3, d_2, d_3 of the party's private key are used during decryption only for the purpose of checking validity of the ciphertext and are not needed to recover the plaintext.

We note that Gentry's scheme also succumbs to the Page-Vercauteren fault attack described in §3.1.

Remark 5. Gentry's scheme (and also the Boyen-Mei-Waters public key encryption scheme) can be modified to resist the fault attack described above. For example, the w component of the ciphertext could be computed as $w = m \oplus G_k(e(g, h_1)^s)$. Here, G_k is chosen from a family $\mathcal{G} = \{G_k\}_{k \in K}$ of keyed hash functions satisfying the so-called entropy smoothing property [30, §3.4], and the range of each G_k is $\{0, 1\}^\ell$ where ℓ is the bitlength of plaintext messages. The description of G_k is included in the PKG's public parameters. Note that plaintext messages are no longer represented as elements in $\mathbb{G}_T$, and so the scheme can be implemented using asymmetric pairings derived from BN curves. It remains to be seen whether this modified scheme enjoys the same provable security properties as Gentry's original scheme.

Remark 6. Kiltz and Vahlis [20] proposed a variant of Gentry's scheme that uses a symmetric-key authenticated encryption scheme E. The fault attack described above will not work on the Kiltz-Vahlis scheme. However, in the Kiltz-Vahlis scheme, a secret key $K \in \mathbb{G}_T$ is generated which is then used *directly* as the key for E. In practice, one needs to use a key derivation function to map K to a bit string of the appropriate length. See, for example, [27, 14] for the use of key derivation functions in the context of (hierarchical) identity-based encryption. It remains to be seen what effect the incorporation of a key derivation function has on the provable security of the scheme.

Remark 7. It can easily be seen that Page-Vercauteren and Whelan-Scott fault attacks are effective on the Boyen-Waters anonymous identity-based encryption schemes [11] (see Appendix A) and the identity-based encryption schemes of Boneh-Boyen [6] and Waters [34] when these protocols are implemented using symmetric pairings. Strictly speaking, the fault attacks lie outside the security models considered in [6], [11] and [34]. These security models only account for indistinguishability of plaintexts against chosen-plaintext attacks and do not permit decryption queries. However, we note that the fault attacks are key recovery attacks and hence go beyond the notion of indistinguishability of plaintexts. Moreover, with

a little extra effort one can mount the fault attack on CCA-secure versions of these schemes that are obtained using the so-called CHK transformation [13].

4.3. **An oblivious transfer protocol.** An adaptive oblivious transfer (AOT) protocol is executed between two parties: a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. The sender $\mathcal{S}$ has a set of secret messages $\mathcal{M} = \{M_1, \ldots, M_N\}$. The receiver $\mathcal{R}$ adaptively obtains messages one at a time in such a way that $\mathcal{S}$ does not learn any information about which messages are accessed while $\mathcal{R}$ does not learn any information about the messages not yet accessed. Camenisch et al. [12] proposed a pairing-based AOT protocol secure in the standard model.

The Camenisch et al. AOT protocol is divided into two stages: (i) Initialization and (ii) Transfer. During Initialization, $\mathcal{S}$ generates some public information, masks the messages in $\mathcal{M}$ using the corresponding secret information, and then sends the masked messages together with the public information to $\mathcal{R}$. Next, $\mathcal{R}$ calls Transfer adaptively. In the j-th round of Transfer, $\mathcal{R}$ chooses a *secret* index $i_j \in \{1, \ldots, N\}$ and executes the protocol with $\mathcal{S}$ to obtain a "secret key" corresponding to the "encryption" of M_{i_j}, thus enabling $\mathcal{R}$ to decrypt the corresponding message. However, $\mathcal{S}$ does not learn any information about i_j while $\mathcal{R}$ learns no information about other elements in $\mathcal{M}$.

4.3.1. *AOT protocol.* The complete description of the Camenisch et al. AOT protocol includes two proofs-of-knowledge (PoK) and a proof-of-membership (PoM). Here we reproduce a slightly simplified version of the AOT protocol; see Figure 2 and Appendix B in [12] for a complete description. The messages M_i are assumed to be elements of $\mathbb{G}_T$.

Initialization.

(i) $\mathcal{S}$ chooses $g, h \in_R \mathbb{G}$, $x \in_R [0, n-1]$, and computes $y = g^x$ and $\alpha = e(g, h)$.
(ii) For each $i \in \{1, \ldots, N\}$, $\mathcal{S}$ computes $A_i = g^{1/(x+i)}$ and $B_i = e(h, A_i) \cdot M_i$. The "ciphertext" corresponding to M_i is $C_i = (A_i, B_i)$.
(iii) Finally, $\mathcal{S}$ sends $(g, y, \alpha, C_1, \ldots, C_n)$ (together with a PoK of h such that $\alpha = e(g, h)$) to $\mathcal{R}$. Note that $\mathcal{S}$ keeps h as its secret key.

Transfer. This protocol is invoked by $\mathcal{R}$ adaptively each time it wants to "decrypt" one of the messages from $\mathcal{M}$. The j-th round of the protocol proceeds as follows:

(i) $\mathcal{R}$ chooses a (secret) index $i_j \in \{1, \ldots, N\}$, $v \in_R [0, n-1]$, and computes $V = A_{i_j}^v$. $\mathcal{R}$ sends V to $\mathcal{S}$ (and a PoK of i_j and v such that $e(V, y) = e(V, g)^{-i_j} \cdot e(g, g)^v$).
(ii) Given V, $\mathcal{S}$ computes

$$W = e(V, h) \tag{8}$$

and sends W to $\mathcal{R}$ (along with a PoM of h such that $\alpha = e(g, h)$ and $W = e(V, h)$).
(iii) Upon receiving W, $\mathcal{R}$ computes $M_{i_j} = B_{i_j} / W^{1/v}$.

Note that $h \in \mathbb{G}$ serves as a "master secret", the knowledge of which allows decryption of *all* messages in $\mathcal{M}$.

4.3.2. *Fault attack.* We assume that the adversary $\mathcal{A}$ can induce a Whelan-Scott sign-change fault (see §3.2) while the pairing value in (8) is being computed. We consider two invocations of TRANSFER where the attacker $\mathcal{A}$ in the role of $\mathcal{R}$ uses the *same* secret index $i^* \in \{1, \ldots, N\}$.

The attack proceeds as follows.

(i) In the first invocation of TRANSFER, $\mathcal{A}$ chooses $v_1 \in_R [0, n-1]$, computes $V_1 = A_{i^*}^{v_1}$, and sends V_1 (along with the PoK of (i^*, v_1)) to $\mathcal{S}$.
(ii) In response, $\mathcal{S}$ is supposed to compute $W_1 = e(V_1, h)$. However, $\mathcal{A}$ induces a sign-change fault while the computation is in progress so that a faulty pairing value $W_1' = e'(V_1, h)$ is computed.
(iii) $\mathcal{S}$ sends W_1' (along with a PoM of h such that $\alpha = e(g, h)$ and $W_1 = e(V_1, h)$) to $\mathcal{A}$.[3] Upon receiving the message, $\mathcal{A}$ terminates the protocol.
(iv) In the second invocation of TRANSFER, $\mathcal{A}$ chooses $v_2 \in_R [0, n-1]$, computes $V_2 = A_{i^*}^{v_2}$, and sends V_2 (along with the PoK of (i^*, v_2)) to $\mathcal{S}$.
(v) In response, $\mathcal{S}$ sends $W_2 = e(V_2, h)$ (together with the PoM). Note that $\mathcal{A}$ allows the protocol to be executed normally, i.e., no fault is applied during the pairing computation.
(vi) $\mathcal{A}$ computes $W_1 = (W_2)^{v_1 v_2^{-1}} = e(V_1, h)$. From W_1 and W_1', $\mathcal{A}$ recovers h as described in §3.2 and thereafter computes all the messages in $\mathcal{M}$.

We note that the AOT protocol also succumbs to the Page-Vercauteren fault attack described in §3.1.

Remark 8. As observed by the authors in [12], the AOT protocol described above can be easily modified to encrypt bitstrings. Thereby, the protocol can be instantiated in the asymmetric pairing setting. In particular, one can use a hash function to map the $\mathbb{G}_T$ element $e(h, A_i)$ to a bitstring which is then XORed with the message (see the computation of B_i in Step (ii) of INITIALIZATION). However, the fault attack still succeeds since $\mathcal{S}$ has to transmit the $\mathbb{G}_T$ element W and a POM of h in Step (ii) of TRANSFER. This is in contrast to the fault attacks on Gentry's IBE and the Boyen-Mei-Waters PKE which can be prevented by using a hash function in the encryption algorithm. To the best of our knowledge, the AOT protocol of [12] is the only example of a pairing-based scheme where there is no apparent countermeasure for the fault attack.

5. Concluding remarks

We have shown that the fault attacks on pairing-based protocols that have been studied in the literature are effective on only a small number of protocols. Most of these protocols require an efficient and reversible method for embedding the message space into the group

[3]The PoM protocol presented in Section B.3 of [12] does not explicitly use $W_1 = e(V_1, h)$. Hence, the PoM protocol terminates even though $\mathcal{S}$ has computed $W_1' = e'(V_1, h)$ instead of W_1. Of course, $\mathcal{A}$ does not care whether the PoM succeeds or not.

$\mathbb{G}_T$. Since such embeddings are only known for certain symmetric pairings, the fault attacks are successful only when the protocols are implemented with these symmetric pairings.

The AOT protocol of [12] is the only example where the fault attack is effective in the asymmetric pairing setting. A useful direction for future work would be to examine the effectiveness of fault attacks on popular identity-based encryption protocols (such as the Boneh-Franklin identity-based encryption scheme) when implemented with asymmetric pairings derived from BN curves.

References

[1] G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for discrete logarithm cryptography". *Pairing-Based Cryptography – Pairing 2013*, LNCS 8365 (2014), 20-44.

[2] G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Weakness of $\mathbb{F}_{3^{6\cdot 1429}}$ and $\mathbb{F}_{2^{4\cdot 3041}}$ for discrete logarithm cryptography", available at http://eprint.iacr.org/2013/737.

[3] G. Adj, A. Menezes, T. Oliveira and F. Rodríguez-Henríquez, "Computing discrete logarithms in $\mathbb{F}_{3^{6\cdot 137}}$ and $\mathbb{F}_{3^{6\cdot 163}}$ using Magma", available at http://eprint.iacr.org/2014/057.

[4] R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, "A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic: Improvements over FFS in small to medium characteristic", *Advances in Cryptology – EUROCRYPT 2014*, LNCS 8441 (2014), 1-16.

[5] P. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order", *Selected Areas in Cryptography – SAC 2005*, LNCS 3897 (2006), 319-331.

[6] D. Boneh and X. Boyen, "Efficient selective identity-based encryption without random oracles", *Journal of Cryptology*, 24 (2011), 659-693.

[7] D. Boneh, R. DeMillo and R. Lipton, "On the importance of checking cryptographic protocols for faults", *Journal of Cryptology*, 14 (2001), 101-119.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", *SIAM Journal on Computing*, 32 (2003), 586-615.

[9] X. Boyen and L. Martin, "Identity-based cryptography standard (IBCS) #1: Supersingular curve implementations of the BF and BB1 cryptosystems", IETF RFC 5091, December 2007.

[10] X. Boyen, Q. Mei and B. Waters, "Direct chosen ciphertext security from identity-based techniques", *12th ACM Conference on Computer and Communications Security – CCS '05*, ACM Press, 320-329.

[11] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)", *Advances in Cryptology – CRYPTO 2006*, LNCS 4117 (2006), 290-307.

[12] J. Camenisch, G. Neven and a. shelat, "Simulatable adaptive oblivious transfer", *Advances in Cryptology – EUROCRYPT 2007*, LNCS 4515 (2007), 573-590.

[13] R. Canetti, S. Halevi and J. Katz, "Chosen-ciphertext security from identity-based encryption", *Advances in Cryptology – EUROCRYPT 2004*, LNCS 3027 (2004), 207-222.

[14] S. Chatterjee and P. Sarkar, "Practical hybrid (hierarchical) identity-based encryption schemes based on the decisional bilinear Diffie-Hellman assumption", *International Journal of Applied Cryptography*, 3 (2013), 47-83.

[15] N. El Mrabet, D. Page and F. Vercauteren, "Fault attacks on pairing-based cryptography", Chapter 13 of *Fault Analysis in Cryptography*, Springer-Verlag, 2012.

[16] C. Gentry, "Practical identity-based encryption without random oracles", *Advances in Cryptology – EUROCRYPT 2006*, LNCS 4004 (206), 445-464.

[17] F. Göloğlu, R. Granger, G. McGuire and J. Zumbrägel, "On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$", *Advances in Cryptology – CRYPTO 2013*, LNCS 8043 (2013), 109-128.

[18] R. Granger, T. Kleinjung and J. Zumbrägel, "Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$)", available at http://eprint.iacr.org/2014/119.
[19] A. Joux, "A new index calculus algorithm with complexity $L(1/4+o(1))$ in very small characteristic", *Selected Areas in Cryptography – SAC 2013*, LNCS 8282 (2014), 355-379.
[20] E. Kiltz and Y. Vahlis, "CCA2 secure IBE: Standard model efficiency through authenticated symmetric encryption", *Topics in Cryptology – CT-RSA 2008*, LNCS 4964 (2008), 221-238.
[21] N. Koblitz and A. Menezes, "Pairing-based cryptography at high security levels", *Cryptography and Coding: 10th IMA International Conference*, LNCS 3796 (2005), 13-36.
[22] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", *Advances in Cryptology – CRYPTO '96*, LNCS 1109 (1996), 104-113.
[23] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis", *Advances in Cryptology – CRYPTO '99*, LNCS 1666 (1999), 388-397.
[24] R. Lashermes, J. Fournier and L. Goubin, "Inverting the final exponentiation of Tate pairings on ordinary elliptic curves using faults", *Cryptographic Hardware and Embedded Systems – CHES 2013*, LNCS 8086 (2013), 365-382.
[25] D. Page and F. Vercauteren, "A fault attack on pairing-based cryptography", *IEEE Transactions on Computers*, 55 (2006), 1075-1080.
[26] K. Rubin and A. Silverberg, "Torus-based cryptography", *Advances in Cryptology – CRYPTO 2003*, LNCS 2729 (2003), 349-365.
[27] P. Sarkar and S. Chatterjee, "Construction of a hybrid HIBE protocol secure against adaptive attacks", *Provable Security – ProvSec 2007*, LNCS 4784 (2007), 51-67.
[28] O. Schirokauer, "The number field sieve for integers of low weight", *Mathematics of Computation*, 79 (2010), 583-602.
[29] M. Scott, "Computing the Tate pairing", *Topics in Cryptology – CT-RSA 2005*, LNCS 3376 (2005), 293-304.
[30] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs", Cryptology ePrint Archive Report 2004/332, available at http://eprint.iacr.org/2013/446.
[31] J. Silverman and J. Tate, *Rational Points on Elliptic Curves*, Springer, 1992.
[32] F. Vercauteren, "The hidden root problem", *Pairing 2008*, LNCS 5209 (2008), 88-99.
[33] E. Verheul, "Evidence that XTR is more secure than supersingular elliptic curve cryptosystems", *Journal of Cryptology*, 17 (2004), 277-296.
[34] B. Waters, "Efficient identity-based encryption without random oracles", *Advances in Cryptology – EUROCRYPT 2005*, LNCS 3494 (2005), 114-127.
[35] C. Whelan and M. Scott, "The importance of the final exponentiation in pairings when considering fault attacks", *Pairing 2007*, LNCS 4575 (2007), 225-246.

Appendix A. Boyen-Waters anonymous identity-based encryption scheme

Boyen and Waters [11] presented an anonymous identity-based encryption scheme and a reductionist security proof that does not invoke the random oracle assumption. Here, 'anonymous' means that the ciphertext does not leak the identity of the intended recipient. The scheme assumes that plaintext messages are elements of $\mathbb{G}_T$.

A.1. **Boyen-Waters scheme.**

Setup. The Private Key Generator (PKG) selects $g, g_0, g_1 \in_R \mathbb{G}$ and $w, t_1, t_2, t_3, t_4 \in_R [0, n-1]$. It computes $\alpha = e(g,g)^{t_1 t_2 w}$ and $v_i = g^{t_i}$ for $i = 1, 2, 3, 4$. The public parameters are $(\alpha, g, g_0, g_1, v_1, v_2, v_3, v_4)$, and the PKG's private key is (w, t_1, t_2, t_3, t_4).

KEY EXTRACTION. To generate a private key for the party with identifier ID $\in [1, n-1]$, the PKG selects $r_1, r_2 \in_R [0, n-1]$ and computes $d_0 = g^{r_1 t_1 t_2 + r_2 t_3 t_4}$, $d_1 = g^{-wt_2}(g_0 g^{\mathrm{ID}})^{-r_1 t_2}$, $d_2 = g^{-wt_1}(g_0 g^{\mathrm{ID}})^{-r_1 t_1}$, $d_3 = (g_0 g_1^{\mathrm{ID}})^{-r_2 t_4}$, and $d_4 = (g_0 g_1^{\mathrm{ID}})^{-r_2 t_3}$. The party's private key is $(d_0, d_1, d_2, d_3, d_4)$.

ENCRYPTION. To encrypt a message $m \in \mathbb{G}_T$ for the party with identifier ID, the sender selects $s, s_1, s_2 \in_R [0, n-1]$ and computes $C' = \alpha^s \cdot m$, $C_0 = (g_0 g_1^{\mathrm{ID}})^s$, $C_1 = v_1^{s-s_1}$, $C_2 = v_2^{s_1}$, $C_3 = v_3^{s-s_2}$, and $C_4 = v_4^{s_2}$. The ciphertext is $C = (C', C_0, C_1, C_2, C_3, C_4)$.

DECRYPTION. To decrypt $C = (C', C_0, C_1, C_2, C_3, C_4)$, the receiver who possesses the private key $(d_0, d_1, d_2, d_3, d_4)$ corresponding to identifier ID computes

$$m = C' \cdot e(C_0, d_0) \cdot e(C_1, d_1) \cdot e(C_2, d_2) \cdot e(C_3, d_3) \cdot e(C_4, d_4). \tag{9}$$

A.2. **Fault attack.** We assume that the adversary $\mathcal{A}$ can induce a Whelan-Scott sign-change fault (see §3.2) while the pairing values in (9) are being computed, and is subsequently able to obtain the decrypted message.

The attack proceeds as follows:

(i) $\mathcal{A}$ selects a plaintext message $m \in \mathbb{G}_T$ and computes the ciphertext $C = (C', C_0, C_1, C_2, C_3, C_4)$.
(ii) $\mathcal{A}$ sends C to the receiver.
(iii) While the receiver computes the pairing value $e(C_0, d_0)$ in (9), $\mathcal{A}$ induces a sign-change fault which causes the receiver to compute the faulty pairing value $e'(C_0, d_0)$.
(iv) $\mathcal{A}$ obtains the (faulty) decryption $m' = C' \cdot e'(C_0, d_0) \cdot e(C_1, d_1) \cdot e(C_2, d_2) \cdot e(C_3, d_3) \cdot e(C_4, d_4)$.
(v) $\mathcal{A}$ computes

$$\frac{m}{m'} = \frac{e(C_0, d_0)}{e'(C_0, d_0)}.$$

The adversary is unable to compute the correct pairing value $e(C_0, d_0)$, but nonetheless is able to narrow the choice for d_0 to at most 12 points; in practice, this number is expected to be 1, 2 or 3 (cf. Example 2).
(vi) Steps (i)–(v) are repeated by inserting faults in the computation of each of the other pairing values in (9), thus yielding a small number of possibilities for each of d_1,d_2, d_3 and d_4.
(vii) For each possible $(d_0, d_1, d_2, d_3, d_4)$, $\mathcal{A}$ decrypts the ciphertext C and checks if the correct plaintext m is obtained; if so, then $\mathcal{A}$ has identified the correct private key with high probability.

Now, with knowledge of $(d_0, d_1, d_2, d_3, d_4)$, the adversary can decrypt any ciphertext that is intended for the party with identifier ID.

We note that the Boyen-Waters scheme also succumbs to the Page-Vercauteren fault attack described in §3.1.

Department of Computer Science and Automation, Indian Institute of Science
E-mail address: sanjit@csa.iisc.ernet.in

Department of Mathematical Sciences, Florida Atlantic University
E-mail address: kkarabina@fau.edu

Department of Combinatorics & Optimization, University of Waterloo
E-mail address: ajmeneze@uwaterloo.ca